

Setting CSS in the Swing SDK Tutorial

Getting Started

Required Skills

The following skills are required prior to working with this tutorial:

- Basic Java programming with the Swing library
- Basic Knowledge of XML

Required Tutorials Completed

The following tutorials are required to be undertaken before attempting this tutorial:

- [Instantiation Tutorial](#)
- [Creating and Editing Configuration Files Tutorial](#)

Tutorial

Step 1. Create a Reference to the External CSS File in the EditLive! Swing SDK Configuration File

To specify an external CSS file for use with the EditLive! Swing SDK, you'll need to make changes to your EditLive! Swing SDK Configuration File. As outlined in the [Creating and Editing Configuration Files Tutorial](#), this is done using a text editor.

Open the *sampleeljconfig.xml* file packaged with EditLive! using a text editor. Locate the following line of code:

```
<!-- <link rel="stylesheet" href="http://www.yourserver.com/style.css" type="text/css" /> -->
```

Remove the <!-- and --> characters wrapping to tag.

```
<link rel="stylesheet" href="http://www.yourserver.com/style.css" type="text/css" />
```

Change the **href** attribute to reference the *main.css* file stored on the Tiny website.

```
<link rel="stylesheet" href="http://www.ephox.com/images/main.css" type="text/css" />
```

Save this configuration file as *styles_config.xml*.

Step 2. Create an Instance of EditLive! for Java in a JFrame

As shown in the [Instantiation Tutorial](#), create an instance of EditLive! for Java in a [JFrame](#).

```

import java.io.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import com.ephox.editlive.*;

public class SetStyles {
    /** html content to appear in the instance of EditLive! for Java */
    private static final String INITIAL_HTML = "<html><head><title>Initial Title</title></head><body><p></p></body></html>";

    /** Base class for EditLive! Swing SDK*/
    private ELJBean editLiveBean = new ELJBean(INITIAL_HTML, "", 700, 570, getClass().getResource("styles_config.xml"), false);

    public SetStyles() throws Exception {
        super("Tutorial - Set Styles");
        this.getContentPane().setLayout(new FlowLayout());

        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                editLiveBean.init();
            }
        });

        // Create a JPanel to hold the ELJBean
        JPanel editorHolder = new JPanel(new FlowLayout());
        editorHolder.add(editLiveBean);
        // Add the JPanel to the JFrame
        this.getContentPane().add(editorHolder);

        // Display the JFrame.
        this.setSize(new Dimension(710, 620));
        this.setVisible(true);

        // Adding a listener to detect if the JFrame is closing, to close the application if needed.
        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        });
    }

    /** Sets up the application and begins its execution
     *
     * @param args the command line arguments - ignored
     */
    public static void main(String args[]) throws Exception {
        try {
            UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
        } catch (Exception e) {
            System.out.println("Unable to locate UIManager class");
            e.printStackTrace();
        }
        new SetStyles();
    }
}

```

Ensure *styles_config.xml* is specified as the EditLive! Swing SDK Configuration File.

Step 3. Specify a String of CSS Using the `setStyles()` Method

EditLive! Swing SDK features a `setStyles()` method that allows developers to specify a string of CSS to be used for the rendering of the editor's HTML content. `setStyles()` allows developers to generate CSS from any location (for example, extracting CSS information stored in a database) and load this CSS into the EditLive! Swing SDK, rather than statically defining the CSS in an EditLive! for Java Configuration File.

```

import java.io.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import com.ephox.editlive.*;

public class SetStyles {
    /** html content to appear in the instance of EditLive! Swing SDK */
    private static final String INITIAL_HTML = "<html><head><title>Initial Title</title></head><body><p></p></body></html>";

    /** Base class for EditLive! for Java */
    private ELJBean editLiveBean = new ELJBean(INITIAL_HTML, "", 700, 570, getClass().getResource("styles_config.xml"), false);

    public SetStyles() throws Exception {
        super("Tutorial - Set Styles");
        this.getContentPane().setLayout(new FlowLayout());

        SwingUtilities.invokeLaterAndWait(new Runnable() {
            public void run() {
                // set CSS for EditLive! and initialize
                editLiveBean.setStyles("h5.warning{color:red;font-weight:bold;}");
                editLiveBean.init();
            }
        });

        // Create a JPanel to hold the ELJBean
        JPanel editorHolder = new JPanel(new FlowLayout());
        editorHolder.add(editLiveBean);
        // Add the JPanel to the JFrame
        this.getContentPane().add(editorHolder);

        // Display the JFrame.
        this.setSize(new Dimension(710, 620));
        this.setVisible(true);

        // Adding a listener to detect if the JFrame is closing, to close the application if needed.
        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        });
    }

    /** Sets up the application and begins its execution
     *
     * @param args the command line arguments - ignored
     */
    public static void main(String args[]) throws Exception {
        try {
            UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
        } catch (Exception e) {
            System.out.println("Unable to locate UIManager class");
            e.printStackTrace();
        }
        new SetStyles();
    }
}

```